IN THE CLAIMS

Please amend the claims as follows:

1-2. (Canceled)

3. (Currently Amended) ~~The host-fabric adapter as claimed in claim 2,~~ A host-fabric adapter installed at a host system for connecting to a switched fabric of a data network, comprising:

a Micro-Engine (ME) arranged to establish connections and support data transfers via said switched fabric;

a serial interface arranged to receive and transmit data packets from said switched fabric for data transfers;

a host interface arranged to receive and transmit host data transfer requests, in the form of descriptors, from said host system for data transfers;

a first-in/first-out (FIFO) interface arranged to receive and transmit data packets to/from said switched fabric via said serial interface, and incorporated therein a Receiver Header Hardware Assist (HWA) Mechanism configured to check header information of incoming data packets host descriptors for header errors so as to offload said Micro-Engine (ME) from having to check for said header errors,

wherein said Receiver Header Hardware Assist (HWA) mechanism comprises:

context registers loaded with context information pertaining to an incoming data packet; header registers loaded with header information of the incoming data packet; and

a processor arranged to execute header checks and comparisons of the header information and the context information and determine whether the incoming data packet is good, and wherein said header checks and comparisons include (1)Version Compare which is a comparison of the context version and the packet version; (2) Priority Compare which is a comparison of the context priority and the packet priority; (3) Source Address Compare which is a comparison of the context destination address and the source address; (4) Source Work Queue Compare which is a comparison of the context destination work queue (WQ) and the source work queue (WQ) number; (5) Port Compare which is a comparison of the context port value and the port the packet was received on; (6) Channel Configuration Check which is a comparison of the OpCode and the context channel configuration type; (7) OpCode Consistency Check which is a comparison of the OpCode with context information; (8) Length Consistency Check which is a comparison of the OpCode with the Length to ensure the length is in the acceptable range for that OpCode; (9) Read Permission Check which is a comparison of the OpCode with the context read permission bit; (10) Write Permission Check which is a comparison of the OpCode with the context write permission bit; (11) Cell Sequence Number Check (CSN) which is a comparison of the received CSN and the context expected CSN; and (12) Packet Sequence Number (PSN) Check which is a comparison of the received PSN with the context expected PSN based to find the relative position of the PSN.

4. (Currently Amended) ~~The host-fabric adapter as claimed in claim 2,~~ A host-fabric adapter installed at a host system for connecting to a switched fabric of a data network, comprising:

a Micro-Engine (ME) arranged to establish connections and support data transfers via said switched fabric;

a serial interface arranged to receive and transmit data packets from said switched fabric for data transfers;

a host interface arranged to receive and transmit host data transfer requests, in the form of descriptors, from said host system for data transfers;

a first-in/first-out (FIFO) interface arranged to receive and transmit data packets to/from said switched fabric via said serial interface, and incorporated therein a Receiver

Header Hardware Assist (HWA) Mechanism configured to check header information of incoming data packets host descriptors for header errors so as to offload said Micro-Engine (ME) from having to check for said header errors,

wherein said Receiver Header Hardware Assist (HWA) mechanism comprises:

context registers loaded with context information pertaining to an incoming data packet; header registers loaded with header information of the incoming data packet; and

a processor arranged to execute header checks and comparisons of the header information and the context information and determine whether the incoming data packet is good, and

wherein said Receiver Header Hardware Assist (HWA) Mechanism first loads the header information of an incoming data packet and the corresponding context information and then processes in parallel all header checks and comparisons at the same time to determine whether the incoming data packet is "good", including compare the context version with the cell/packet version, compare the context priority with the cell/packet priority, compare the context destination address with the source address of the incoming cell/packet, compare the context destination work queue (WQ) number with the source work queue (WQ) number of the incoming cell/packet, compare the context port value with the port the incoming cell/packet was received, check for channel configuration error, check for OpCode consistency, check for length consistency, check for read permission, check for write permission, check for Cell Sequence Number (CSN) and check for Packet Sequence Number (PSN).

5. (Original) The host-fabric adapter as claimed in claim 4, wherein said Receiver FIFO Hardware Assist (HWA) mechanism may be implemented as an Application Specific Integrated Circuit (ASIC).

6. (Original) The host-fabric adapter as claimed in claim 5, wherein said processor of the Receiver FIFO Hardware Assist (HWA) mechanism comprises:

different sets of header comparators arranged to process in parallel all header checks and comparisons; and

a combine logic arranged to indicate whether an incoming data packet is good based on header check results.

7. (Original) The host-fabric adapter as claimed in claim 6, wherein said combine logic corresponds to an AND gate which responds to all header check results and, if all those header check results are successful, generates an indication that the incoming data packet is "good", and alternatively, generates an indication that the incoming data packet is "bad" if any of those header check results is unsuccessful.

8. (Original) The host-fabric adapter as claimed in claim 6, wherein said processor further comprises:

an Error Status Register connected to output lines of the header comparators to register as error status bits if any one of those header check results is unsuccessful; and

a Multiplexer arranged to produce ME readable data to enable said Micro-Engine (ME) to determine the error status registered.

9. (Original) The host-fabric adapter as claimed in claim 6, wherein said header comparators comprise a series of Compare Logics, including XOR gates and AND gates arranged in parallel to make comparisons between the context version with the cell/packet version of an incoming cell/packet, the context priority with the cell/packet priority, the context destination address with the source address of the incoming cell/packet, the context destination work queue (WQ) number with the source work queue (WQ) number of the incoming cell/packet, the context port value with the port the incoming cell/packet was received, and check for channel configuration error, read permission, write permission, and Cell Sequence Number (CSN).

10. (Original) The host-fabric adapter as claimed in claim 6, wherein one set of header comparators includes a Packet Sequence Number (PSN) Compare Logic configured to find the relative position of a PSN of an incoming data packet with respect to an expected PSN (ePSN).

11. (Original) The host-fabric adapter as claimed in claim 10, wherein said PSN Compare Logic comprises:

a first PSN comparator arranged to compare the ePSN from the incoming data packet and the context PSN (cPSN) from the context information and determine whether the cPSN equals to the ePSN;

a second PSN comparator arranged to compare the ePSN which has included a constant **X** (total # of PSNs)/2, and the cPSN to determine whether the cPSN is greater than or equals to a Start of Earlier Range (SER);

a third PSN comparator arranged to compare the ePSN and the cPSN and determine whether the cPSN is less than the ePSN;

a fourth PSN comparator arranged to compare the ePSN and the cPSN and determine whether the ePSN is greater than a constant **Y** ((total # of PSNs)/2-1); and

a combine logic arranged to receive PSN comparisons and generate three outputs, including a PSN Earlier, a PSN Later, and a PSN Equal.

12. (Original) The host-fabric adapter as claimed in claim 11, wherein said first to fourth PSN comparators correspond to XOR gates, and said combine logic comprises:

a first AND gate arranged to logically combine outputs of the second, third, and fourth PSN comparators;

a first OR gate arranged to logically combine outputs of the third and fourth PSN comparators;

a second AND gate arranged to logically combine an inverted output of the fourth PSN comparator and an output of the first OR gate;

a second OR gate arranged to receive outputs of the first and second AND gates;

a third AND gate arranged to receive an inverted output of the first PSN comparator and an inverted output of the second OR gate and produce the PSN After; and

a fourth AND gate arranged to receive an inverted output of the first PSN comparator and an output of the second OR gate and produce the PSN Early.

13. (Canceled)

14. (Currently Amended) ~~The host-fabric adapter as claimed in claim 13,~~ A host-fabric adapter installed at a host system for connecting to a switched fabric of a data network, comprising:

a Micro-Engine (ME) arranged to establish connections and support data transfers via said switched fabric;

a serial interface arranged to receive and transmit data packets from said switched fabric for data transfers;

a host interface arranged to receive and transmit host data transfer requests, in the form of descriptors, from said host system for data transfers;

a first-in/first-out (FIFO) interface arranged to receive and transmit data packets to/from said switched fabric via said serial interface, and incorporated therein a Receiver Header Hardware Assist (HWA) Mechanism configured to check header information of incoming data packets host descriptors for header errors so as to offload said Micro-Engine (ME) from having to check for said header errors;

an address translation interface which provides an interface for address translation, and which is addressable by write data and system controls from said Micro-Engine (ME), via a system data bus and a system control bus;

a context memory which provides an interface to a context manager, and which is addressable by write data and system controls from said Micro-Engine (ME), via said system data bus and said system control bus, for providing the necessary context for a work queue pair used for sending and receiving data packets;

a local bus interface which provides an interface to a local bus, and which is addressable by write data and system controls from said Micro-Engine (ME), via said system data bus and said system control bus, for supporting system accessible context connections and data transfers; and

a completion queue/doorbell manager interface which provides an interface to completion queues, and doorbell and memory registration rules, and which is addressable by write data and system controls from said Micro-Engine (ME), via said system data bus and said system control bus;

wherein said local bus interface incorporates therein a Transmitter Header Hardware Assist (HWA) Mechanism configured to generate OpCode and Length fields for an outgoing packet when an entire packet is being assembled for transmission, via the serial interface so as to offload said Micro-Engine (ME) from MicroCode processing.

15. (Original) The host-fabric adapter as claimed in claim 14, wherein said Transmitter Header Hardware Assist (HWA) Mechanism is configured to compute the OpCode and Length fields of a data packet simultaneously using context information and descriptors from a host system.

16. (Original) The host-fabric adapter as claimed in claim 15, wherein said Transmitter Header Hardware Assist (HWA) Mechanism comprises:

context registers loaded with context information pertaining to a packet to be assemble for transmission, via the serial interface; and

a processor arranged to determine the OpCode and Length fields of a data packet based on the context information, the maximum call size information, and control information from a descriptors posted.

17. (Original) The host-fabric adapter as claimed in claim 16, wherein said processor comprises logic gates and a look-up table which take the inputs and perform the following

functions: (1) Computes Payload Length; (2) Computes Number of Bytes other than Payload; (3) Computes the OpCode; and (4) Store results in task associated registers selectable by said Micro-Engine (ME).

18. (Original) The host-fabric adapter as claimed in claim 15, wherein said Transmitter Header Hardware Assist (HWA) Mechanism computes the OpCode of a data packet by:

loading work queue (WQ) status information from a WP status register, and the packet bytes remaining to transmit;

determining whether packet bytes remaining to transmit are greater than a maximum packet size based on a maximum transfer size;

when the packet bytes remaining to transmit are greater than the maximum packet size, determining whether there is a message in progress;

if there is no message in progress, indicating the packet bytes remain as a first packet;

if there is a message in progress, indicating the packet bytes remain as a middle packet;

when the packet bytes remaining to transmit are not greater than the maximum packet size, also determining whether there is a message in progress;

if there is no message in progress, indicating the packet bytes remain as the only packet;

if there is a message in progress, indicating the packet bytes remain as a last packet; and

determining if a Read, Write, or Send request with or without Immediate Data is associated with the packet bytes remain, and generating the OpCode based on the Read, Write, or Send request with or without Immediate Data associated with the packet bytes.

19. (Original) The host-fabric adapter as claimed in claim 18, wherein said Transmitter Header Hardware Assist (HWA) Mechanism computes the Length of a data packet simultaneously with computation of the OpCode by:

determining if the packet bytes remaining to transmit are greater than a maximum transfer size, after the packet bytes remaining to transmit are loaded;

if the packet bytes remaining to transmit are greater than the maximum transfer size, indicating that the Length equals to the maximum transfer size; and

if the packet bytes remaining to transmit are not greater than the maximum transfer size, indicating that the Length equals to the packet bytes remain.

20. (Original) The host-fabric adapter as claimed in claim 19, wherein said OpCode and Length fields of a data packet are subsequently loaded into a packet buffer for packet construction with other header information before said data packet is scheduled for transmission, via the serial interface.

21. (Currently Amended) The host-fabric adapter as claimed in claim [[13]] 14, wherein said Micro-Engine (ME) comprises:

one or more Data Multiplexers arranged to supply appropriate interface data based on an ME instruction;

an Instruction Memory arranged to provide said ME instruction based on downloadable microcode;

an Arithmetic Logic Unit (ALU) arranged to perform mathematical, logical and shifting operations, and supply write data to the host interface, the address translation interface, the context memory interface, the local bus interface, the completion queue/doorbell manager interface, the Receive FIFO interface and the Transmit FIFO interface, via said system write data bus; and

an Instruction Decoder arranged to supply system controls to the host interface, the address translation interface, the context memory interface, the local bus interface, the completion queue/doorbell manager interface, the Receive FIFO interface and the Transmit FIFO interface, via said system control bus, to execute said ME instruction from said Instruction

Memory to control operations of said Data Multiplexers, and to determine functions of said Arithmetic Logic Unit (ALU).

22. (Original) The host-fabric adapter as claimed in claim 21, wherein said Instruction Memory corresponds to a static random-access-memory (SRAM) provided to store MicroCode that are downloadable for providing said ME instruction to said Instruction Decoder.

23. (Canceled)

24. (Original) A host-fabric adapter installed at a host system for connecting to a switched fabric of a data network, comprising:

a Micro-Engine (ME) arranged to establish connections and support data transfers via said switched fabric;

a serial interface arranged to receive and transmit data packets from said switched fabric for data transfers;

a host interface arranged to receive and transmit host data transfer requests, in the form of descriptors, from said host system for data transfers;

a first-in/first-out (FIFO) interface arranged to receive and transmit data packets to/from said switched fabric via said serial interface; and

a Transmitter Header Hardware Assist (HWA) Mechanism configured to generate OpCode and Length fields for an outgoing data packet when an entire data packet is being assembled for transmission, via the serial interface so as to offload said Micro-Engine (ME) from having to build all data packets for data transfers.

25. (Original) The host-fabric adapter as claimed in claim 24, wherein said Transmitter Header Hardware Assist (HWA) Mechanism computes the OpCode and Length fields of a data packet simultaneously using context information and descriptors from a host system.

26. (Original) The host-fabric adapter as claimed in claim 25, wherein said Transmitter Header Hardware Assist (HWA) Mechanism comprises:

context registers loaded with context information pertaining to a packet to be assemble for transmission, via the serial interface; and

a processor arranged to determine the OpCode and Length fields of a data packet based on the context information, the maximum call size information, and control information from a descriptors posted.

27. (Original) The host-fabric adapter as claimed in claim 26, wherein said processor comprises logic gates and a look-up table which take the inputs and perform the following functions: (1) Computes Payload Length; (2) Computes Number of Bytes other than Payload; (3) Computes the OpCode; and (4) Store results in task associated registers selectable by said Micro-Engine (ME).

28. (Original) The host-fabric adapter as claimed in claim 25, wherein said Transmitter Header Hardware Assist (HWA) Mechanism computes the OpCode of a data packet by:

loading work queue (WQ) status information from a WP status register, and the packet bytes remaining to transmit, via the serial interface;

determining whether packet bytes remaining to transmit are greater than a maximum packet size based on a maximum transfer size;

when the packet bytes remaining to transmit are greater than the maximum packet size, determining whether there is a message in progress;

if there is no message in progress, indicating the packet bytes remain as a first packet;

if there is a message in progress, indicating the packet bytes remain as a middle packet;

when the packet bytes remaining to transmit are not greater than the maximum packet size, also determining whether there is a message in progress;

if there is no message in progress, indicating the packet bytes remain as the only packet;

if there is a message in progress, indicating the packet bytes remain as a last packet; and

determining if a Read, Write, or Send request with or without Immediate Data is associated with the packet bytes remain, and generating the OpCode based on the Read, Write, or Send request with or without Immediate Data associated with the packet bytes.

29. (Original) The host-fabric adapter as claimed in claim 28, wherein said Transmitter Header Hardware Assist (HWA) Mechanism computes the Length of a data packet simultaneously with computation of the OpCode by:

determining if the packet bytes remaining to transmit are greater than a maximum transfer size, after the packet bytes remaining to transmit are loaded;

if the packet bytes remaining to transmit are greater than the maximum transfer size, indicating that the Length equals to the maximum transfer size; and

if the packet bytes remaining to transmit are not greater than the maximum transfer size, indicating that the Length equals to the packet bytes remain.

30. (Original) The host-fabric adapter as claimed in claim 29, wherein said OpCode and Length fields of a data packet are subsequently loaded into a packet buffer for packet construction with other header information before said data packet is scheduled for transmission, via the serial interface.

31. (Original) The host-fabric adapter as claimed in claim 24, wherein said host interface, said serial interface and said Micro-Engine (ME) are configured in accordance with the "*Virtual Interface (VI) Architecture Specification*", the "*Next Generation Input/Output (NGIO) Specification*" and the "*InfiniBand^{TM} Specification*".

32. (Original) A host-fabric adapter, comprising:

a Micro-Engine (ME) arranged to establish connections and support data transfers via a switched fabric;

a serial interface arranged to receive and transmit data packets from said switched fabric for data transfers;

a host interface arranged to receive and transmit host data transfer requests, in the form of descriptors, from said host system for data transfers;

a Receiver Header Hardware Assist (HWA) Mechanism configured to check header information of incoming data packets host descriptors for header errors so as to offload said Micro-Engine (ME) from having to check for said header errors; and a Transmitter Header Hardware Assist (HWA) Mechanism configured to generate OpCode and Length fields for an outgoing data packet when an entire data packet is being assembled for transmission, via the serial interface, so as to offload said Micro-Engine (ME) from having to build all data packets for data transfers.

33. (Original) The host-fabric adapter as claimed in claim 32, wherein said Receiver Header Hardware Assist (HWA) Mechanism loads the header information of an incoming data packet and the corresponding context information and then processes in parallel all header checks and comparisons at the same time to determine whether the incoming data packet is "good", including compare the context version with the cell/packet version, compare the context priority with the cell/packet priority, compare the context destination address with the source address of the incoming cell/packet, compare the context destination work queue (WQ) number with the source work queue (WQ) number of the incoming cell/packet, compare the context port value with the port the incoming cell/packet was received, check for channel configuration error, check for OpCode consistency, check for Length consistency, check for read permission, check for write permission, check for Cell Sequence Number (CSN) and check for Packet Sequence Number (PSN).

34. (Original) The host-fabric adapter as claimed in claim 33, wherein said Receiver Header Hardware Assist (HWA) mechanism comprises:

context registers loaded with context information pertaining to an incoming data packet;

header registers loaded with header information of the incoming data packet; and

a processor arranged to execute header checks and comparisons of the header information and the context information and determine whether the incoming data packet is "good".

35. (Original) The host-fabric adapter as claimed in claim 34, wherein said processor of the Receiver FIFO Hardware Assist (HWA) mechanism comprises:

different sets of header comparators arranged to process in parallel all header checks and comparisons; and a combine logic arranged to indicate whether an incoming data packet is good based on header check results.

36. (Currently Amended) The host-fabric adapter as claimed in claim [[6]] 35, wherein said combine logic corresponds to an AND gate which responds to all header check results and, if all those header check results are successful, generates an indication that the incoming data packet is "good", and alternatively, generates an indicate that the incoming data packet is "bad" if any of those header check results is unsuccessful.

37. (Original) The host-fabric adapter as claimed in claim 36, wherein said processor further comprises:

an Error Status Register connected to output lines of the header comparators to register as error status bits if any one of those header check results is unsuccessful; and

a Multiplexer arranged to produce ME readable data to enable said Micro-Engine (ME) to determine the error status registered.

38. (Original) The host-fabric adapter as claimed in claim 35, wherein one set of header comparators includes a Packet Sequence Number (PSN) Compare Logic configured to find the relative position of a PSN of an incoming data packet with respect to an expected PSN (ePSN).

39. (Original) The host-fabric adapter as claimed in claim 38, wherein said PSN Compare Logic comprises:

a first PSN comparator arranged to compare the ePSN from the incoming data packet and the context PSN (cPSN) from the context information and determine whether the cPSN equals to the ePSN;

a second PSN comparator arranged to compare the ePSN which has included a constant **X** (total # of PSNs)/2, and the cPSN to determine whether the cPSN is greater than or equals to a Start of Earlier Range (SER);

a third PSN comparator arranged to compare the ePSN and the cPSN and determine whether the cPSN is less than the ePSN;

a fourth PSN comparator arranged to compare the ePSN and the cPSN and determine whether the ePSN is greater than a constant **Y** ((total # of PSNs)/2-1); and

a combine logic arranged to receive PSN comparisons and generate three outputs, including a PSN Earlier, a PSN Later, and a PSN Equal.

40. (Original) The host-fabric adapter as claimed in claim 39, wherein said first to fourth PSN comparators correspond to XOR gates, and said combine logic comprises:

a first AND gate arranged to logically combine outputs of the second, third, and fourth PSN comparators;

a first OR gate arranged to logically combine outputs of the third and fourth PSN comparators;

a second AND gate arranged to logically combine an inverted output of the fourth PSN comparator and an output of the first OR gate;

a second OR gate arranged to receive outputs of the first and second AND gates;

a third AND gate arranged to receive an inverted output of the first PSN comparator and an inverted output of the second OR gate and produce the PSN After; and

a fourth AND gate arranged to receive an inverted output of the first PSN comparator and an output of the second OR gate and produce the PSN Early.

41. (Original) The host-fabric adapter as claimed in claim 32, wherein said Transmitter Header Hardware Assist (HWA) Mechanism is configured to compute the OpCode and Length fields of a data packet simultaneously using context information and descriptors from a host system.

42. (Original) The host-fabric adapter as claimed in claim 41, wherein said Transmitter Header Hardware Assist (HWA) Mechanism comprises:

context registers loaded with context information pertaining to a packet to be assemble for transmission, via the serial interface; and a processor arranged to determine the OpCode and Length fields of a data packet based on the context information, the maximum call size information, and control information from a descriptors posted.

43. (Original) The host-fabric adapter as claimed in claim 42, wherein said processor comprises logic gates and a look-up table which take the inputs and perform the following functions: (1) Computes Payload Length; (2) Computes Number of Bytes other than Payload; (3) Computes the OpCode; and (4) Store results in task associated registers selectable by said Micro-Engine (ME).

44. (Original) The host-fabric adapter as claimed in claim 32, wherein said Transmitter Header Hardware Assist (HWA) Mechanism computes the OpCode of a data packet by:

loading work queue (WQ) status information from a WP status register, and the packet bytes remaining to transmit;

determining whether packet bytes remaining to transmit are greater than a maximum packet size based on a maximum transfer size;

when the packet bytes remaining to transmit are greater than the maximum packet size, determining whether there is a message in progress;

if there is no message in progress, indicating the packet bytes remain as a first packet;

if there is a message in progress, indicating the packet bytes remain as a middle packet;

when the packet bytes remaining to transmit are not greater than the maximum packet size, also determining whether there is a message in progress;

if there is no message in progress, indicating the packet bytes remain as the only packet;

if there is a message in progress, indicating the packet bytes remain as a last packet; and

determining if a Read, Write, or Send request with or without Immediate Data is associated with the packet bytes remain, and generating the OpCode based on the Read, Write, or Send request with or without Immediate Data associated with the packet bytes.

45. (Original) The host-fabric adapter as claimed in claim 44, wherein said Transmitter Header Hardware Assist (HWA) Mechanism computes the Length of a data packet simultaneously with computation of the OpCode by:

determining if the packet bytes remaining to transmit are greater than a maximum transfer size, after the packet bytes remaining to transmit are loaded;

if the packet bytes remaining to transmit are greater than the maximum transfer size, indicating that the Length equals to the maximum transfer size; and

if the packet bytes remaining to transmit are not greater than the maximum transfer size, indicating that the Length equals to the packet bytes remain.

46. (Original) The host-fabric adapter as claimed in claim 45, wherein said OpCode and Length fields of a data packet are subsequently loaded into a packet buffer for packet construction with other header information before said data packet is scheduled for transmission, via the serial interface.

47. (Original) The host-fabric adapter as claimed in claim 32, wherein said host interface, said serial interface and said Micro-Engine (ME) are configured in accordance with the "*Virtual Interface (VI) Architecture Specification*", the *"Next Generation Input/Output (NGIO) Specification"* and the *"InfiniBand^{TM} Specification"*.